Claims 6, 15, 44, and 53 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over *Idicula* in view of *Deinhart* and *Jensenworth,*, and further in view of U.S. Patent No. 6,233,576 issued to Lewis ("*Lewis*"). This rejection is respectfully traversed.

Claims 12 and 50 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over *Idicula* in view of *Deinhart* and *Jensenworth*, and further in view of Official Notice. This rejection is respectfully traversed.

II.   ISSUES RELATING TO THE CITED ART

Claims 1-4, 7-11, 13, 16-20, 39-42, 45-49, 51, and 54-58 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Idicula* in view of *Deinhart*, and further in view of *Jensenworth*.

A.   CLAIM 1

Claim 1 recites:

> A computer-implemented method for controlling access to a resource of a plurality of resources, the method comprising the steps of:
> creating and storing in a filesystem of an Operating System a plurality of files that each represents a different resource of the plurality of resources;
> assigning an access value to a file attribute of a file that represents the resource, wherein the file attribute is used by the Operating System to manage file access, wherein the access value corresponds to a combination of a particular role and the resource;
> receiving user-identifying information from a user requesting access to the resource, wherein the user-identifying information comprises a role associated with the user, **wherein the role is determined from a user identifier uniquely associated with the user and from a group identifier associated with a group that includes the user;**
> **receiving a resource identifier associated with the resource;**
> **creating an access identifier based on the user-identifying information and the resource identifier, wherein the access identifier is formatted as a file attribute that is used by the Operating System to manage file access;**
> **calling the Operating System to perform a file operation on the file, wherein calling the Operating System includes providing the access identifier to the Operating System; and**
> **granting the user access to the resource only when the Operating System call successfully performs the file operation, wherein the Operating**

System call successfully performs the file operation if the access identifier matches the access value;
> wherein the file operation on the file representing the resource is selected from a group consisting of opening the file, closing the file, deleting the file, reading from the file, writing to the file, executing the file, appending to the file, reading a file attribute, and writing a file attribute. (emphasis added)

1.   *Discussion of Claim 1*

In an embodiment, resources may be a software application, software application message, server, router, switch, file, directory, etc. (Specification, paragraphs 27 and 28). A file is created and stored in a filesystem of an Operating System. The file represents a particular resource of a plurality of resource. An access value is assigned to a file attribute of that file. The file attribute is used by the Operating System to manage file access. The access value corresponds to a combination of a particular role and the resource.

Subsequently, user-identifying information is received from a user who is requesting access to that resource. A resource ID associated with the resource is also received, e.g., from the user. Subsequently, an access ID is created based on the user-identifying information and the resource ID. The access ID is formatted as a file attribute that is used by the Operating System to gain access to the file. The Operating System is then called to perform a file operation (e.g., open, write, delete) on the file that represents the resource. This is done by providing the access ID to the Operating System. The user is granted access to the particular resource when the Operating System call successfully performs the file operation. The Operating System call successfully performs the file operation if the access identifier matches the access value.

Because Operating System calls are used to attempt access to a file, a complex security mechanism is avoided and application software developers are not required to learn a specific API and to write code against it (see specification, paragraph 3). Claim 1 uses the filesystem of an Operating System (OS) to manage access to files. The OS filesystem is one of the most well-

known and easy to write code systems in a computer system (see specification, paragraphs 4, 7, and 8).

> 2.   Jensenworth *fails to teach or suggest that the recited access value corresponds to a particular role and a resource*

Claim 1 recites "assigning an access value to a file attribute of a file that represents the resource, wherein the file attribute is used by the Operating System to manage file access, wherein the access value corresponds to a combination of a particular role and the resource." This access value is later used in conjunction with the recited access identifier that is created after a user requests access to a particular resource. The Operating System call successfully performs a file operation on the file that represents the particular resource if the access identifier matches the access value.

The Final Office Action cites col. 5, lines 4-21 of *Jensenworth* for allegedly disclosing this step of Claim 1. This is incorrect. Presumably, the Final Office Action equates the security descriptor 76 or access control entries of *Jensenworth* with the recited access value of Claim 1. However, *Jensenworth* fails to even mention the term "role." Consequently, *Jensenworth* cannot teach or suggest that the security descriptor 76 or an access control entry corresponds to a combination of a particular role and a resource, as Claim 1 recites.

> 3.   Deinhart *fails to teach or suggest the recited role*

The Final Office Action cites col. 1, lines 31-36 of *Deinhart* for allegedly disclosing "wherein the role is determined from a user identifier uniquely associated with the user and from a group identifier associated with a group that includes the user" as recited in Claim 1. This is incorrect. That cited portion of *Deinhart* merely states that "access rights are granted or revoked explicitly for individual users or groups of users on respective data or, more generally, on respective objects by a system administrator." This portion of *Deinhart* is completely unrelated

to how a role is determined. Specifically, this cited portion of *Deinhart* fails to teach or suggest

that a role is determined from <u>two</u> <u>specific</u> identifiers: a user identifier and a group identifier.

   4.   Idicula *fails to teach or suggest the recited resource identifier*

Claim 1 further recites: "receiving a resource identifier associated with the resource."

The Final Office Action cites col. 7, lines 19-35 of *Idicula* for allegedly disclosing this step of

Claim 1. This is incorrect. The applicable part of that cited portion merely states:

> In step 220, a request is received from a client that starts a session. For example, a
> request is received from database client 102a for database services. The database
> server determines whether a session has already been created for this client by
> checking the contents of process state object 130. <u>If a session is already created</u>
> <u>for this client, a session object 122 associated with the client is indicated in the</u>
> <u>process state object 130</u>. (emphasis added)

It is unclear what in *Idicula* the Final Office Action is equating to the recited resource identifier

and the recited resource. If the Final Office Action is equating the session object 122 of *Idicula*

with the recited resource, then it is unclear in what sense a "resource identifier associated with

the [session object, i.e., the alleged resource]" is received. Furthermore, as Claim 1 further

recites, an access identifier is created based on user-identifying information <u>and the resource</u>

<u>identifier</u>. Nothing in *Idicula* is created based on an identifier of a session object (i.e., the

alleged resource identifier). No other reasonable interpretation of *Idicula* provides the claimed

resource identifier.

   5.   Idicula *fails to teach or suggest the recited access identifier*

The Final Office Action cites col. 4, lines 42-56 in *Idicula* for allegedly disclosing

"creating an access identifier based on the user-identifying information and the resource

identifier, wherein the access identifier is formatted as a file attribute that is used by the

Operating System to manage file access" as recited in Claim 1. This is incorrect. That portion

merely states:

> The database server 110 includes memory 120 on the database server host
> computer, which is allocated for use by the database server 110. The database
> server 110 maintains several data structures in memory 120. Among the data
> structures maintained in memory 120 are zero or more <u>session objects 122, one or</u>
> <u>more process state objects 130a, 130b, collectively referenced hereinafter as</u>
> <u>process state objects 130, and a session pool object 140. In object-oriented</u>
> <u>technologies, an object is a data structure that stores data that indicates one or</u>
> <u>more attributes or methods or both</u>. An object is one instance of an object type
> that is defined in one or more object type data structures. In other embodiments,
> the data structures that are illustrated as objects in FIG. 1 need not be objects
> according to object-oriented technologies. (emphasis added)

This cited portion refers to several data structures that are maintained in memory of a database

server. In the claims, the recited access identifier is created based on user-identifying

information and a resource identifier, but the cited portion of *Idicula* fails to refer to *any*

identifiers. Furthermore, this step of Claim 1 recites that the access identifier is formatted as a

file attribute that is used by an Operating System to manage file access. This cited portion of

Idicula fails to mention anything remotely related to how an identifier is formatted and an

Operating System.

> 6. Idicula *fails to teach or suggest calling an Operating System to perform a*
> *file operation on a file*

The Final Office Action cites col. 1, lines 52-62 and col. 7, lines 19-30 of *Idicula* for

allegedly disclosing "calling the Operating System to perform a file operation on the file,

wherein calling the Operating System includes providing the access identifier to the Operating

System" as recited in Claim 1. This is incorrect. Nowhere does *Idicula* teach or suggest that an

___operating system___ **is called in the manner recited**. The **only** portion of *Idicula* that refers to an

operating system is the following:

> A session is a related series of one or more requests for services made over a
> communication channel. The channel is typically established by the operating
> system of the host for the database server and that persists for one or more
> communications from the client, depending on the communications protocol used
> by the client. (emphasis added)

Thus, the **only** function the operating system of *Idicula* performs is the establishing of a

communication channel between a database client and a database server. However, the operating

system in *Idicula* is not called to perform a file operation. Therefore, it is not surprising that

Idicula must fail to teach or suggest that calling an Operating System includes providing an

access identifier to the Operating System.

> 7.   Idicula *fails to teach or suggest granting a user access to the resource
>   only when the Operating System call successfully performs a file operation*

The Final Office Action cites col. 7, lines 20-21 of *Idicula* for allegedly disclosing

"granting the user access to the resource only when the Operating System call successfully

performs the file operation" as recited in Claim 1. This is incorrect. As indicated above, the

only mention of an operating system in *Idicula* is regarding the establishment of a

communication channel between a database client and a database server. The single mention of

Idicula's operating system is in **no way related to** the granting of access to a resource.

Therefore, *Idicula* fails to teach or suggest this step of Claim 1.

Based on the foregoing, the cited art fails to teach or suggest, either individually or in

combination, all the limitations of Claim 1. Therefore, Claim 1 is patentable over the cited art.

Reconsideration and withdrawal of the rejection of Claim 1 under 35 U.S.C. § 103(a) is therefore

respectfully requested.

B.   CLAIMS 10, 18, 39, 48, AND 56

The Final Office Action stated the same reasons in rejecting Claims 10 and 18 to those in rejecting present Claim 1. Also, Claims 39, 48 and 56 recite features discussed above that make Claim 1 patentable over the cited art. Therefore, for at least the same reasons set forth above by in connection with present Claim 1, it is respectfully submitted that each of Claims 10, 18, 39, 48 and 56 is patentable over the cited art.

C.   DEPENDENT CLAIMS

The dependent claims not discussed thus far are dependent claims, each of which depends (directly or indirectly) on one of the independent claims discussed above. Each of the dependent claims is therefore allowable for the reasons given above for the claim on which it depends. In addition, each of the dependent claims introduces one or more additional limitations that may independently render it patentable.

1.   Claims 7, 16, 45, and 54

For example, the Final Office Action asserts that Claims 7, 16, 45, and 54 do <u>not</u>:

> carry patentable weight since the claim recites the file operation of 'opening the file representing the resource,' which was optionally recited in Claims 1, 10, 18, 22, 31, 39, 48, and 56…, upon which the said respective claims depend. Therefore, since the opening of the file is optional and not necessary to the claimed invention, the claim is rejected. (pages 5-6)

This rationale is inconsistent with well established patent practice principles. To review, Claim 1 recites that the file operation could be any of the recited operations. Thus, if there is prior art that anticipates or renders unpatentable all the other features of Claim 1, the prior art would only have to additionally teach or suggest that the recited file operation is one of the recited operations. Claim 7 depends on Claim 1 and further limits Claim 1 by reciting, among other things, the step of opening the recited file. Therefore, the opening of the file is <u>no longer</u>

optional in Claim 7. Therefore, the Final Office Action's premise that the opening of the file is optional is incorrect. Furthermore, the Final Office Action has failed to cite any references for teaching or suggesting the steps of Claim 7 that are not found in Claim 1.

Due to the fundamental differences already identified, to expedite the positive resolution of this case, a separate discussion of those limitations is not included at this time. The Applicants reserve the right to further point out the differences between the cited art and the novel features recited in the dependent claims.

III. CONCLUSIONS & MISCELLANEOUS

For the reasons set forth above, all of the pending claims are now in condition for allowance. The Examiner is respectfully requested to contact the undersigned by telephone relating to any issue that would advance examination of the present application.

A petition for extension of time, to the extent necessary to make this reply timely filed, is hereby made. If applicable, a law firm check for the petition for extension of time fee is enclosed herewith. If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to charge any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: October 27, 2008

/DanielDLedesma#57181/
Daniel D. Ledesma
Reg. No. 57,181

2055 Gateway Place Suite 550
San Jose, California 95110-1083
Telephone No.: (408) 414-1080 ext. 229
Facsimile No.: (408) 414-1076